

# Introducing Dash for R



---

*Ryan Patrick Kyle, Senior R Developer* • R-Ladies Montréal, 19 September 2019

- What's Dash?
- A brief tour of the Dash gallery
- Installing Dash for R + libraries
- Building a simple Dash app
- Helpful resources
- Q&A

# What's Dash?



- Framework for creating analytical web applications in Python and R
- Created by Chris Parmer, originally released in 2017
- Dash for Python relies on Flask as the underlying web server, while Dash for R uses Fierly and httpuv
- Make functional, sophisticated apps — *no JavaScript required*
- Dash for R is MIT-licensed, open source, and completely free!

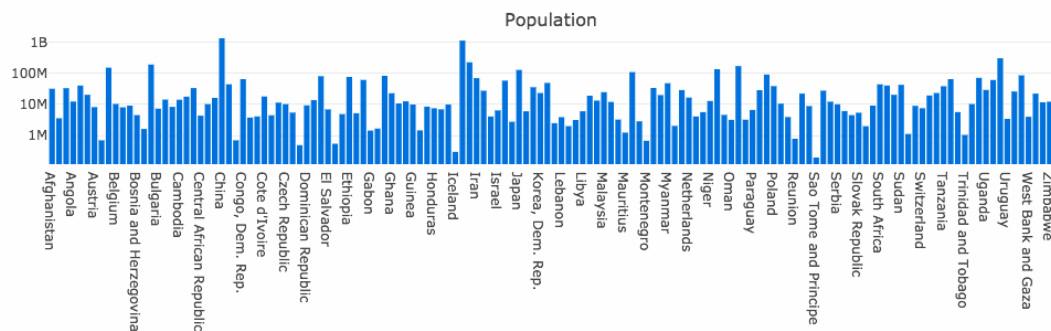
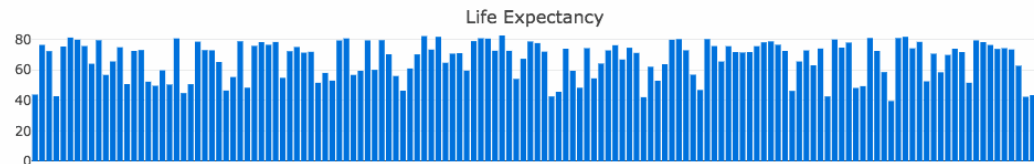
# What's Dash?



- Dash for R development began in summer 2017
- Early work by Carson Sievert
- Officially released in July during useR! 2019
- Syntax nearly identical to Dash for Python, by design
- Interface elements are provided by React components

## Dash DataTable

| FILTER ROWS              |           |             |                  |         |           |      |
|--------------------------|-----------|-------------|------------------|---------|-----------|------|
| <input type="checkbox"/> | continent | country     | gdpPercap        | lifeExp | pop       | year |
| <input type="checkbox"/> | Asia      | Afghanistan | 974.5803384      | 43.828  | 31889923  | 2007 |
| <input type="checkbox"/> | Europe    | Albania     | 5937.02952599999 | 76.423  | 3600523   | 2007 |
| <input type="checkbox"/> | Africa    | Algeria     | 6223.367465      | 72.301  | 33333216  | 2007 |
| <input type="checkbox"/> | Africa    | Angola      | 4797.231267      | 42.731  | 12420476  | 2007 |
| <input type="checkbox"/> | Americas  | Argentina   | 12779.3796400000 | 75.32   | 40301927  | 2007 |
| <input type="checkbox"/> | Oceania   | Australia   | 34435.3674399999 | 81.235  | 20434176  | 2007 |
| <input type="checkbox"/> | Europe    | Austria     | 36126.4927       | 79.829  | 8199783   | 2007 |
| <input type="checkbox"/> | Asia      | Bahrain     | 29796.0483399999 | 75.635  | 708573    | 2007 |
| <input type="checkbox"/> | Asia      | Bangladesh  | 1391.253792      | 64.062  | 150448339 | 2007 |



## Dash Component Libraries

- **Dash Core Components**
- **Dash HTML Components**
- **Dash Table**

- Dash Bio
- Dash Cytoscape
- Dash Canvas
- Dash DAQ

- *Dash Design Kit\**
- *Dash Snapshot Engine\**
- *Dash Analytics\**

Let's take a look at a few sample applications ...

# Dash Gallery



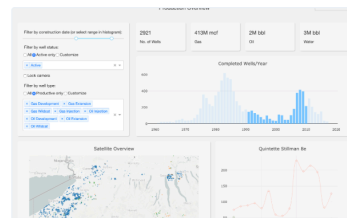
plotly | Dash

## Dash App Gallery

Click on a demo app's name for more info and links to [Python and R source code](#) where available. More info at <https://plot.ly/dash>

All Apps (45)

Search applications...



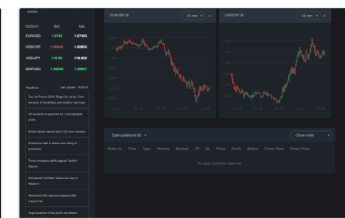
New York Oil and Gas

Energy Cross-filtering



Manufacturing SPC Dashboard

Data Acquisition Streaming



FOREX Web Trader

Streaming Finance



Financial Report

Finance



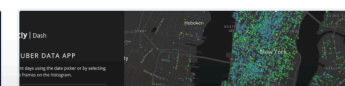
Multipage Report

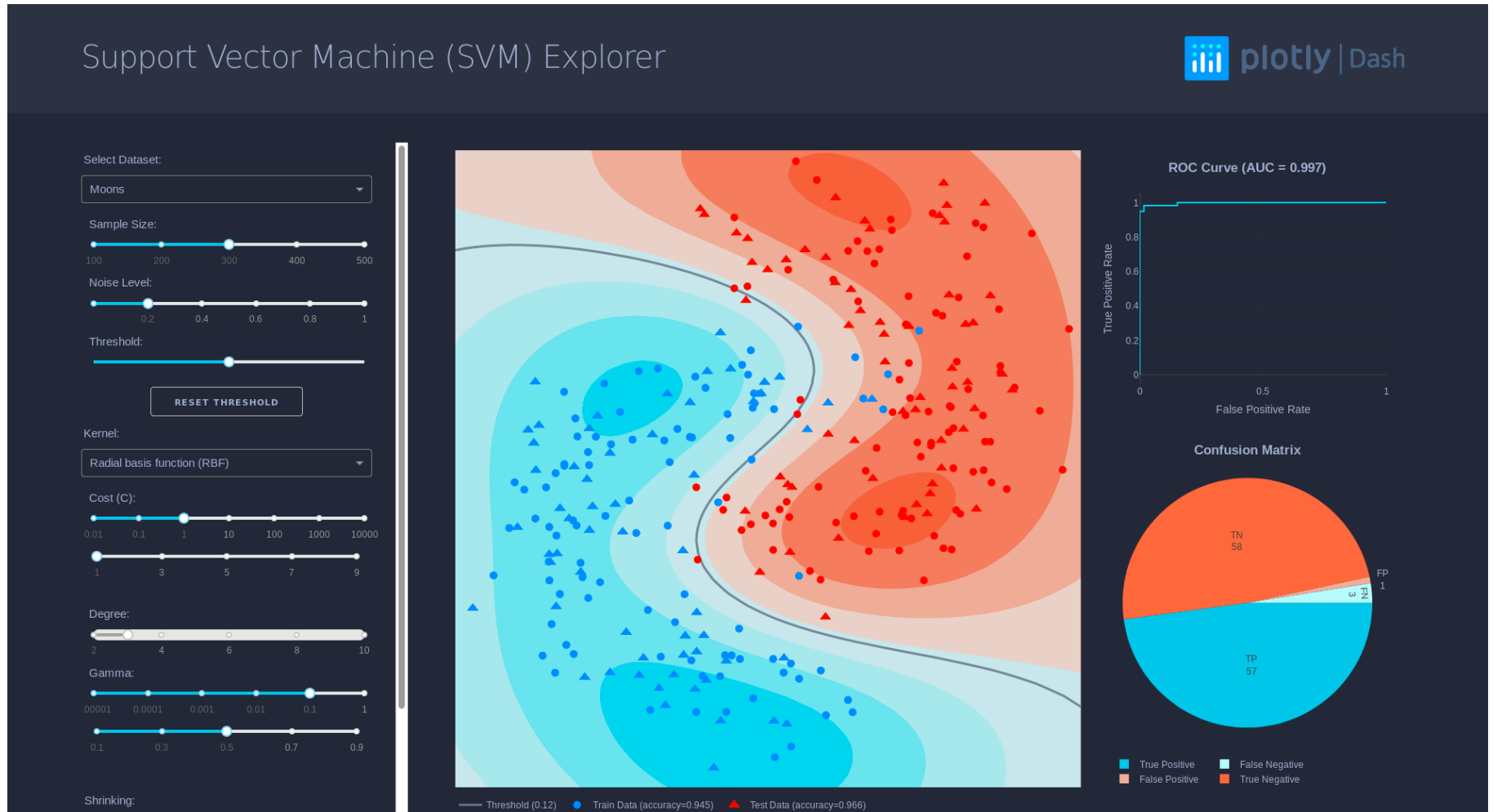
Reporting Finance



Support Vector Machine

Machine Learning







# Dash Gallery



dash

NCBI Nucleotide DB Explorer

About

Explorer Controls

Enter a single GenBank accession ID or multiple ID's separated by commas and select the button to generate an alignment or the sequence of the dataset. Enter additional datasets to add these sequences to the alignment.

Example: Single Dataset - NR\_108049

Multiple Datasets - JF806202, HM161150, FJ356743, JF806202, JQ073190, GU457971, FJ356741, JF806

GU457971, FJ356741, JF806

GENERATE SEQUENCE

GENERATE ALIGNMENT

DOWNLOAD FASTA DATA

Drag and drop FASTA files or select files.

Multiple Sequence Alignment Chart

Search for a Dataset:

Use keywords to search the Nucleotide database for a gene or organism of interest. This will retrieve the top 10 related datasets' Accession IDs along with their descriptions.

Example searches: "Basiliscus basiliscus[Organism]" or "BRCA1[Gene]"

Select the table cell containing your desired Accession ID to add this dataset to the alignment and generate a new figure. Then, filter through datasets to analyze the CpG patterns and nucleotide compositions for your sequence of interest.

Enter a species or gene...

SEARCH FOR DATASETS

Sequence Viewer

998

Search in sequence..1

1

TCCATGCTCT

CGAAGTCT

GAAGATGCT

TCTTGCAG

TTACAT

91

CAGAGTAC

CGATATGAT

GGGTTTGA

GGACAGCA

TTGGTCT

181

ATGCTGCT

GGTTCGCT

TGATGAGT

TGAGAGCA

CATCTAT

271

CTTCACTG

TGTGATTA

GATGCTGT

ATGAGATG

AGATGTT

361

TTGATGCT

TTTCACTG

ATGAGTCT

CTGTCACT

TGCAAT

451

TGTGTTGT

ACCTTTGT

CTATGCTG

CTGATGAT

AGACAT

541

AGGCTATG

AGAGTGT

CTGATCTT

ATATGCTG

AACTGCT

631

AGCTGCTG

TGAGTATG

GGCTTATG

CTTCACT

TACTAT

721

ACCTATCT

TCTTCACT

ACAGAGAT

ATGAGATG

CTTATG

811

ATGAGTCT

TGAGAGT

AGAGGTTT

CTGCAAGC

TTTATG

901

ATGAGTCT

ATTTTATG

ATATTCAT

TGAGATG

TGAGAT

991

AGTCACT

CpG Island Distribution

Nucleotide Base Composition

Selected Dataset

JF806202

HM161150

FJ356743

JQ073190

GU457971

FJ356741

# Dash Gallery



## Rate of US Poison-Induced Deaths

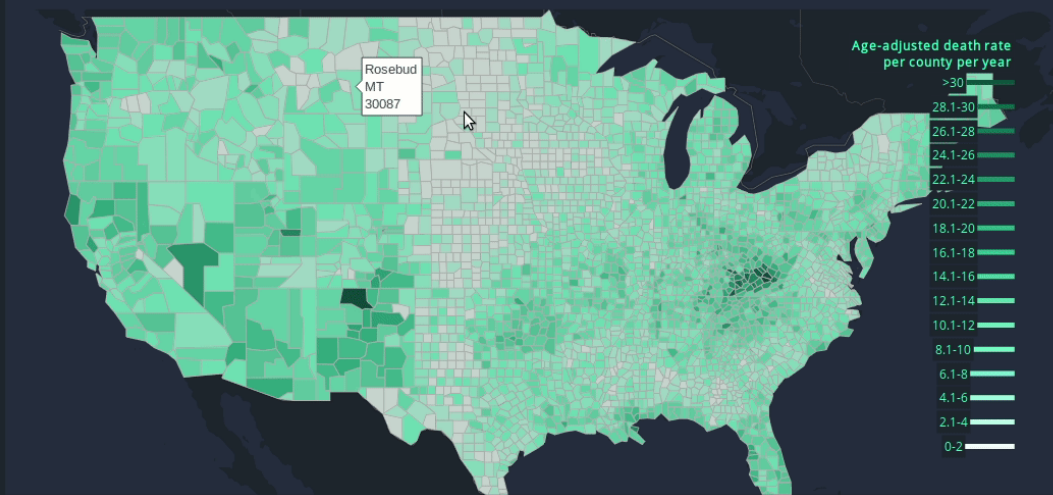


† Deaths are classified using the International Classification of Diseases, Tenth Revision (ICD-10). Drug-poisoning deaths are defined as having ICD-10 underlying cause-of-death codes X40-X44 (unintentional), X60-X64 (suicide), X85 (homicide), or Y10-Y14 (undetermined intent).

Drag the slider to change the year:



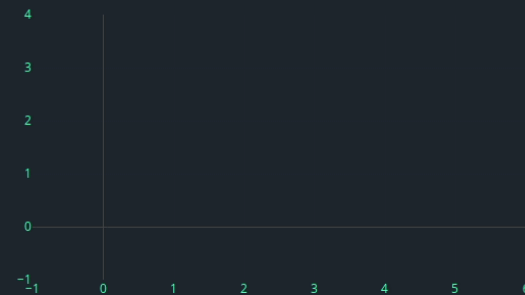
Heatmap of age adjusted mortality rates from poisonings in year 2004



Select chart:

Age-adjusted death rate (single year)

Click-drag on the map to select counties



*Cool! How do I get started using Dash?*

# Installing Dash



- Version 1.0 of the **dash** package will land on CRAN soon
- For now, the current release is available via GitHub!

```
library(devtools)  
# installs dashHtmlComponents, dashCoreComponents, and dashTable  
install_github("plotly/dashR", upgrade = TRUE)
```

- Install syntax is very similar for other Dash packages:

```
install_github("plotly/dash-bio", upgrade = TRUE)  
install_github("plotly/dash-canvas", upgrade = TRUE)  
install_github("plotly/dash-cytoscape", upgrade = TRUE)  
install_github("plotly/dash-daq", upgrade = TRUE)
```

# Building a simple Dash app



- Dash apps have two fundamental parts:
  - **Layout** - *provides structure, controls app appearance*
  - **Callbacks** - *update component properties whenever inputs change*
- If you've used Shiny, the **layout** portion of a Dash app is analogous to the **ui** of a Shiny app
- And a callback is (vaguely) like **server** in a Shiny app

- Dash apps can have multiple **callback** statements
- Dash for R also supports multiple **outputs**
- The **layout** portion of a Dash app is a hierarchical tree of page elements —in Dash, we can nest components as **children** of other components

# Building a simple Dash app



- Breaking it down a bit more, as I see it:
  - In R, components are functions whose arguments are component properties (in Python, they are classes)
  - Callbacks produce **outputs** from inputs, which update component properties (e.g. values, figures, hoverdata)
  - This relationship is *very general*, once you've got the basic idea, the rest is pretty straightforward! 😊



# Building a simple Dash app

---



- OK, let's try building a Dash for R app from scratch
- I'll individually demo the app features one-by-one
- ... then tie them together in the last step

Brief break for questions / live coding exercise

Brief break for questions / live coding exercise

*if you'd like to follow along, just visit*

<https://github.com/rpkyle/rladies-mtl-demo>

# Helpful Resources



# Helpful Resources

---



- <https://dashr.plot.ly> (Interactive R documentation)
- <https://community.plot.ly/> (Community forum)
- Several component libraries also provide vignettes
- <https://dash-gallery.plotly.host/Portal/> (Sample apps)

# Helpful Resources

---



- <https://twitter.com/plotlygraphs>
- [ryan@plot.ly](mailto:ryan@plot.ly)
- <https://twitter.com/ryanpkyle>

# Acknowledgements



- Dash Development Team and the rest of my talented colleagues at Plotly
- Our stellar crew of interns in the summer of 2019
- And to R-Ladies MTL for giving me a chance to present